Toxicity & Safety Pipeline — Initial Implementation

🧭 Overview:

We're building a modular pipeline to evaluate the **toxicity and safety profile** of small-molecule drug candidates. Your job is to implement a **first working version** using a curated set of state-of-the-art models.

This pipeline will:

- Flag safety concerns early in the drug development process
- Produce a composite toxicity score
- Surface interpretability outputs (e.g., risk breakdown, confidence)

📥 Inputs & 📤 Outputs

🔹 Inputs:

- Required: **SMILES string**
- Optional: Mockable inputs for image-based or 3D models (e.g., histopathology, CT, descriptors)

🔹 Output JSON:

```
{
  "composite_score": 0.71,
  "organ_toxicity": {
    "cardiotoxicity": 0.64,
    "hepatotoxicity": 0.81
  },
  "neurotoxicity": 0.42,
  "mitochondrial_toxicity": 0.55,
  "tissue_accumulation": {
    "liver": "high",
    "brain": "moderate"
  },
  "morphological_cytotoxicity": 0.68,
  "immunotoxicity": 0.33,
```

```
  "structural_alerts": [],
  "ld50": 310,
  "flags": ["high hepatotoxicity", "morphological concern"],
  "model_confidence": 0.92
}
```

🔧 Modules to Implement

Each module should be in its own script or function. Use mocks or placeholder values if needed — we'll swap in models later.

---

1. Input Preprocessing

- Convert SMILES → RDKit molecule
- Compute molecular descriptors (MACCS, ECFP)
- Optional: generate 3D conformers if required later

---

2. Structural Alerts

- Use RDKit filters (PAINS, BRENK)
- Output: list of triggered alerts + alert count

---

3. General Toxicity

- Model: **TDC-2**
  - Pull LD50, carcinogenicity, and general tox predictions
  - Return normalized 0–1 toxicity score

---

4. Organ-Specific Toxicity

- Models:
  - **H-optimus-0** (tissue-level pathology)
  - **UNI** (rare damage patterns, subtle signs)
  - **Merlin** (CT-based preclinical toxicity — stub this)

- Output: Per-organ risk (liver, kidney, heart, etc.)

---

5. Neurotoxicity

- Model: **CONVERGE**
    - Predict CNS toxicity
    - Stub with a 0–1 output for now

---

6. Mitochondrial Toxicity

- Model: MITO-Tox or stubbed classifier
    - Return mito risk probability

---

7. Tissue Accumulation

- Model: pkCSM or mock
    - Return per-organ accumulation: “low”, “moderate”, or “high”

---

8. Morphological Cytotoxicity

- Model: **IMPA**
    - Predicts perturbation-induced changes in cell morphology
    - Return 0–1 score

---

9. Immunotoxicity (New)

- Stubbed for now
    - Could later use immune-related data from TDC-2 or NetMHCpan-style outputs
    - Output: 0–1 immunotoxicity risk

---

10. Explainability & Confidence (New)

- Return per-module confidence if supported (e.g., std dev of ensemble, calibration curve)

- Track which modules disagreed (e.g., H-optimus vs. UNI)

---

11. Scoring & Aggregation

- Combine module scores into a final composite toxicity score (0–1)

```
score = (
    0.15 * general_tox +
    0.2 * organ_tox_avg +
    0.15 * neurotox +
    0.1 * mito_tox +
    0.1 * morpho_tox +
    0.1 * accumulation_penalty +
    0.1 * immunotox +
    0.1 * structural_alert_penalty
)
```

- Normalize and threshold where needed
- Add flags for high-risk values (e.g., if any individual risk > 0.8)

✅ What Success Looks Like

- You input a SMILES string and return a complete, valid JSON
- Each module is its own function/class
- Clear stubs or placeholder values are used where real models aren't yet integrated
- Logging is clean and readable (i.e., no print spam)
- JSON includes model confidence, flags, and clean structure

🛠️ Implementation Notes

- Organize code under:
  `/tox_pipeline/modules/` for models
  `/tox_pipeline/utils/` for helpers
  `/tox_pipeline/run_pipeline.py` as the main entry script

- You can mock model outputs with simple random numbers (e.g., `np.random.uniform(0,1)`) as placeholders
- Add TODO comments where real model integration will go